US010437540B2

(12) **United States Patent**
Mastie et al.

(10) **Patent No.: US 10,437,540 B2**
(45) **Date of Patent: *Oct. 8, 2019**

(54) **WHITE SPACE MANAGEMENT MECHANISM**

(71) Applicant: **Ricoh Company, Ltd.**, Tokyo (JP)

(72) Inventors: **Scott D. Mastie**, Longmont, CO (US); **Craig D. Brossman**, Durango, CO (US)

(73) Assignee: **Ricoh Company, Ltd.**, Tokyo (JP)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **16/164,485**

(22) Filed: **Oct. 18, 2018**

(65) **Prior Publication Data**

US 2019/0050185 A1 Feb. 14, 2019

Related U.S. Application Data

(63) Continuation of application No. 15/665,087, filed on Jul. 31, 2017, now Pat. No. 10,140,075, which is a continuation of application No. 12/771,005, filed on Apr. 30, 2010, now Pat. No. 9,760,328.

(51) **Int. Cl.**
G06F 3/12 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 3/1288*** (2013.01); ***G06F 3/1219*** (2013.01); ***G06F 3/1244*** (2013.01)

(58) **Field of Classification Search**
USPC 358/3.31, 3.32, 1.12–1.18, 3.23, 1.2, 1.5, 358/1.9, 537–540
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,513,254	A	4/1996	Markowitz	
7,190,474	B1	3/2007	Silverbrook et al.	
7,403,302	B2 *	7/2008	Gann	H04N 1/00204 358/1.15
8,072,635	B2 *	12/2011	Roberts	G06Q 10/00 358/1.15
2002/0165937	A1	11/2002	Nitta et al.	
2003/0071988	A1	9/2003	Sugihara	
2004/0115396	A1	1/2004	Satomi	
2004/0059435	A1	3/2004	Goldberg et al.	
2004/0095587	A1	5/2004	Brown et al.	
2004/0158655	A1	8/2004	Kremer	
2004/0215559	A1	10/2004	Rebenack	
2006/0206380	A1	9/2006	Joo	
2007/0115498	A1	5/2007	Noel et al.	
2007/0288843	A1	12/2007	Makino	
2008/0137132	A1	6/2008	Perronnin	

(Continued)

OTHER PUBLICATIONS

PCT Search Report and Written Opinion, PCT/US2011/033123, dated Jul. 27, 2011, 7 pages.

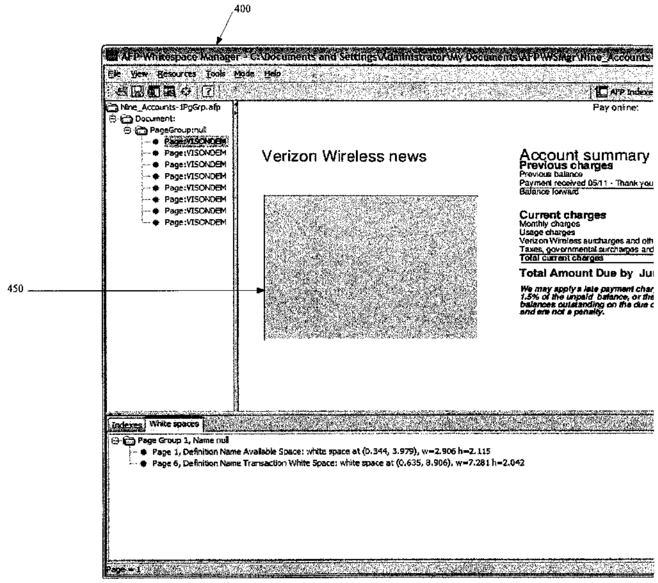
(Continued)

Primary Examiner — Marcellus J Augustin
(74) *Attorney, Agent, or Firm* — Jaffery Watson Mendonsa & Hamilton LLP

(57) **ABSTRACT**

A computer generated method disclosed. The method includes examining a print job data stream spooled in a spooled print production file, identifying one or more white space regions within documents in the print production file and inserting content into each of the white space regions.

20 Claims, 12 Drawing Sheets



AFP Whitespace Manager - C:\Documents and Settings\Administrator\My Documents\AFP WSMgr\Nine_Accounts

Nine_Accounts-1PgGrp.afp
Document:
PageGroup:null
Page:VISONDEM
Page:VISONDEM
Page:VISONDEM
Page:VISONDEM
Page:VISONDEM
Page:VISONDEM
Page:VISONDEM
Page:VISONDEM

Pay online:

Verizon Wireless news

Account summary
Previous charges
Previous balance
Payment received 05/11 - Thank you
Balance forward

Current charges
Monthly charges
Usage charges
Verizon Wireless surcharges and oth
Taxes, governmental surcharges and
Total current charges

Total Amount Due by Ju

We may apply a late payment char 1.5% of the unpaid balance, or the balances outstanding on the due c and are not a penalty.

Page Group 1, Name null
Page 1, Definition Name Available Space: white space at (0.344, 3.979), w=2.906 h=2.115
Page 6, Definition Name Transaction White Space: white space at (0.635, 8.906), w=7.281 h=2.042

Page = 1

(56) **References Cited**

U.S. PATENT DOCUMENTS

2008/0291486	A1	11/2008	Isles et al.
2008/0313035	A1	12/2008	Peeters
2009/0039583	A1	2/2009	Horn et al.

OTHER PUBLICATIONS

Non-Final Office Action from U.S. Appl. No. 12/770,997 dated Aug. 29, 2012. 11 pages.
Response to Non-Final Office Action as filed Oct. 4, 2012 for U.S. Appl. No. 12/770,997, 9 pages.
Notice of Allowance from U.S. Appl. No. 15/665,087 dated Jul. 24, 2018, 10 pages.

* cited by examiner

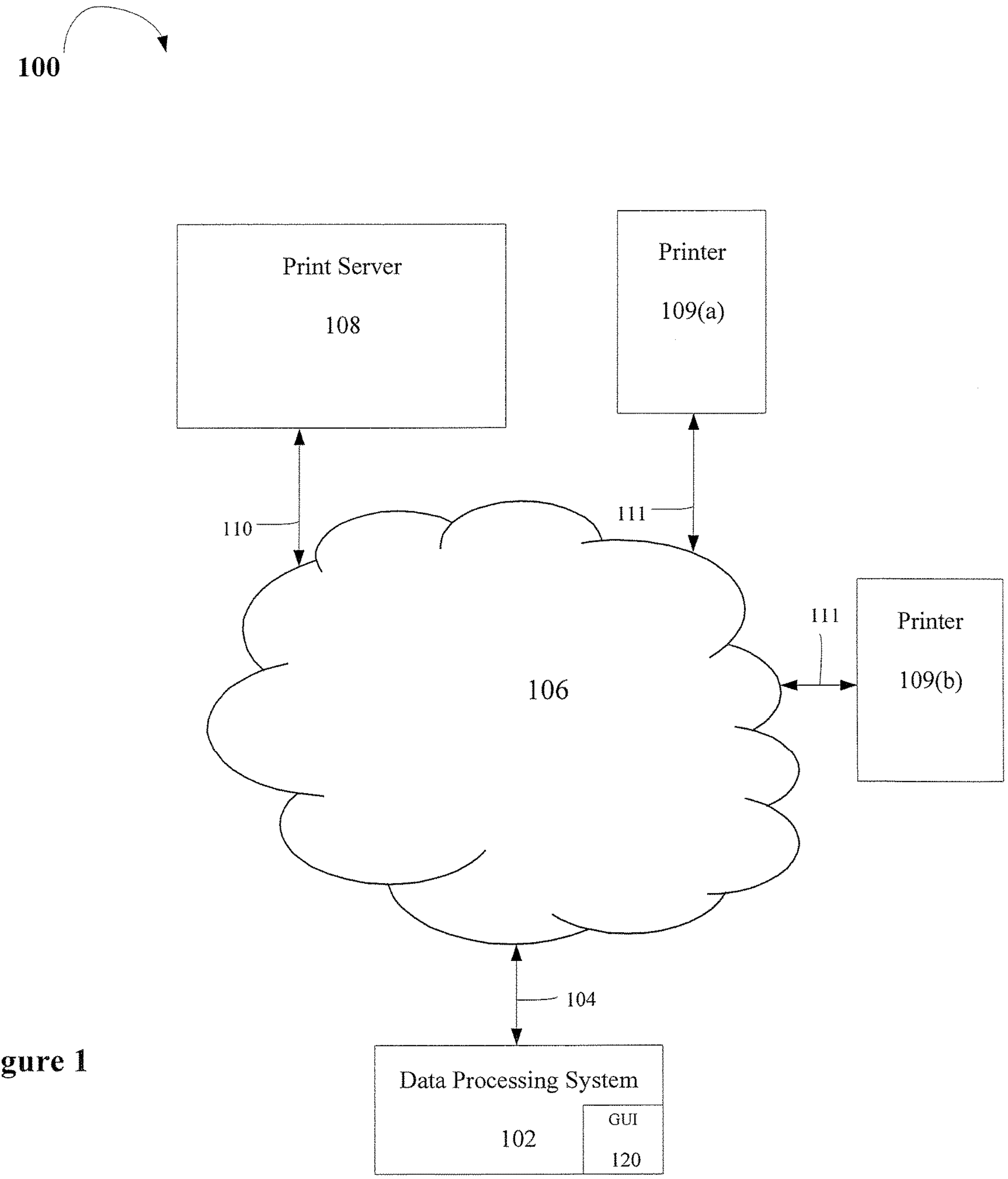

100
Print Server
108
Printer
109(a)
110
111
106
111
Printer
109(b)
104
Data Processing System
102
GUI
120

Figure 1

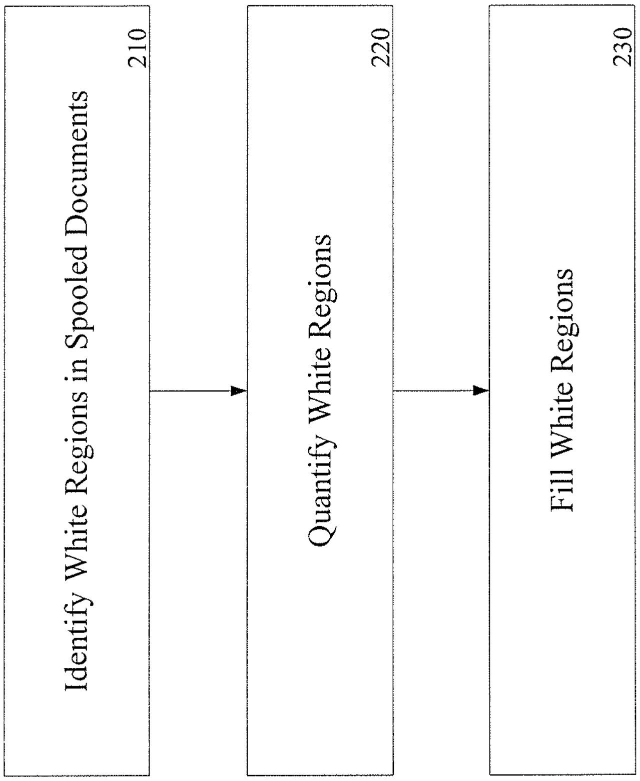

Identify White Regions in Spooled Documents
210
Quantify White Regions
220
Fill White Regions
230

Figure 2

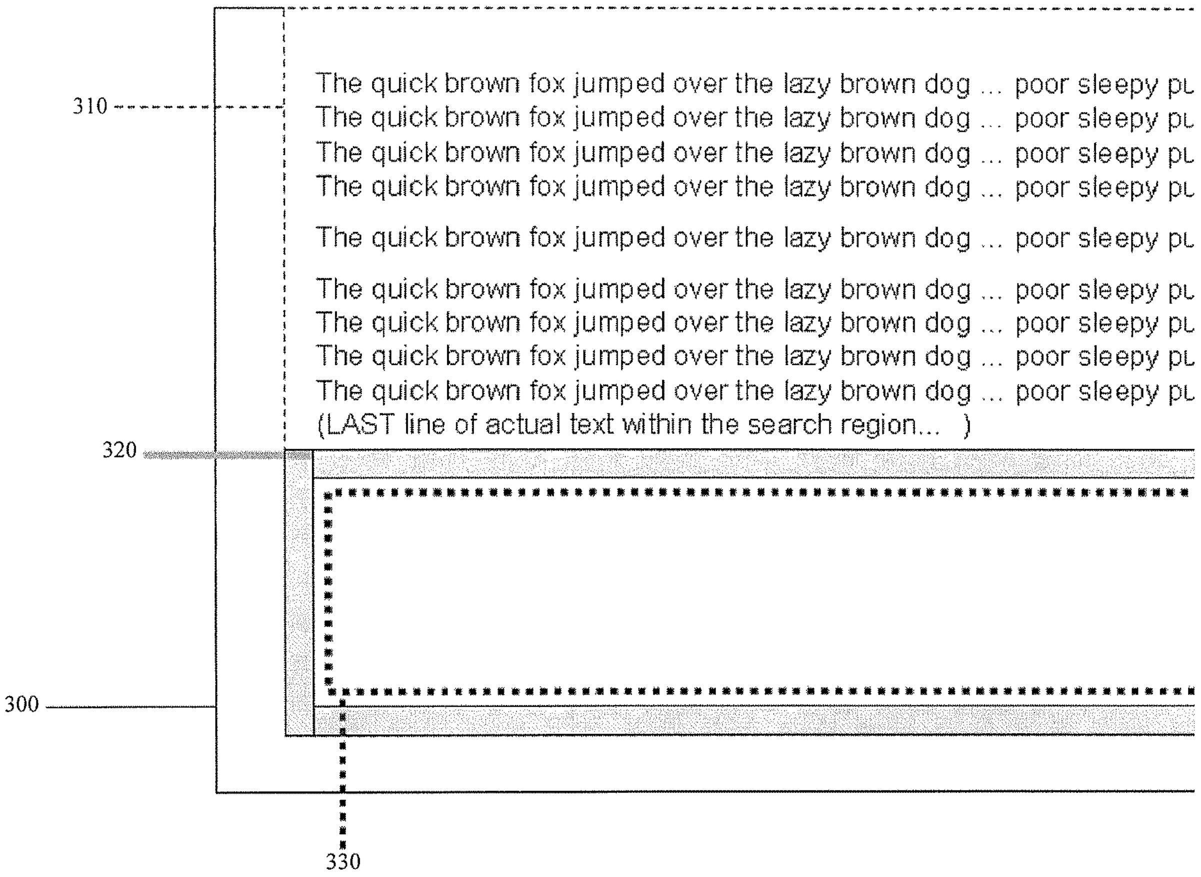
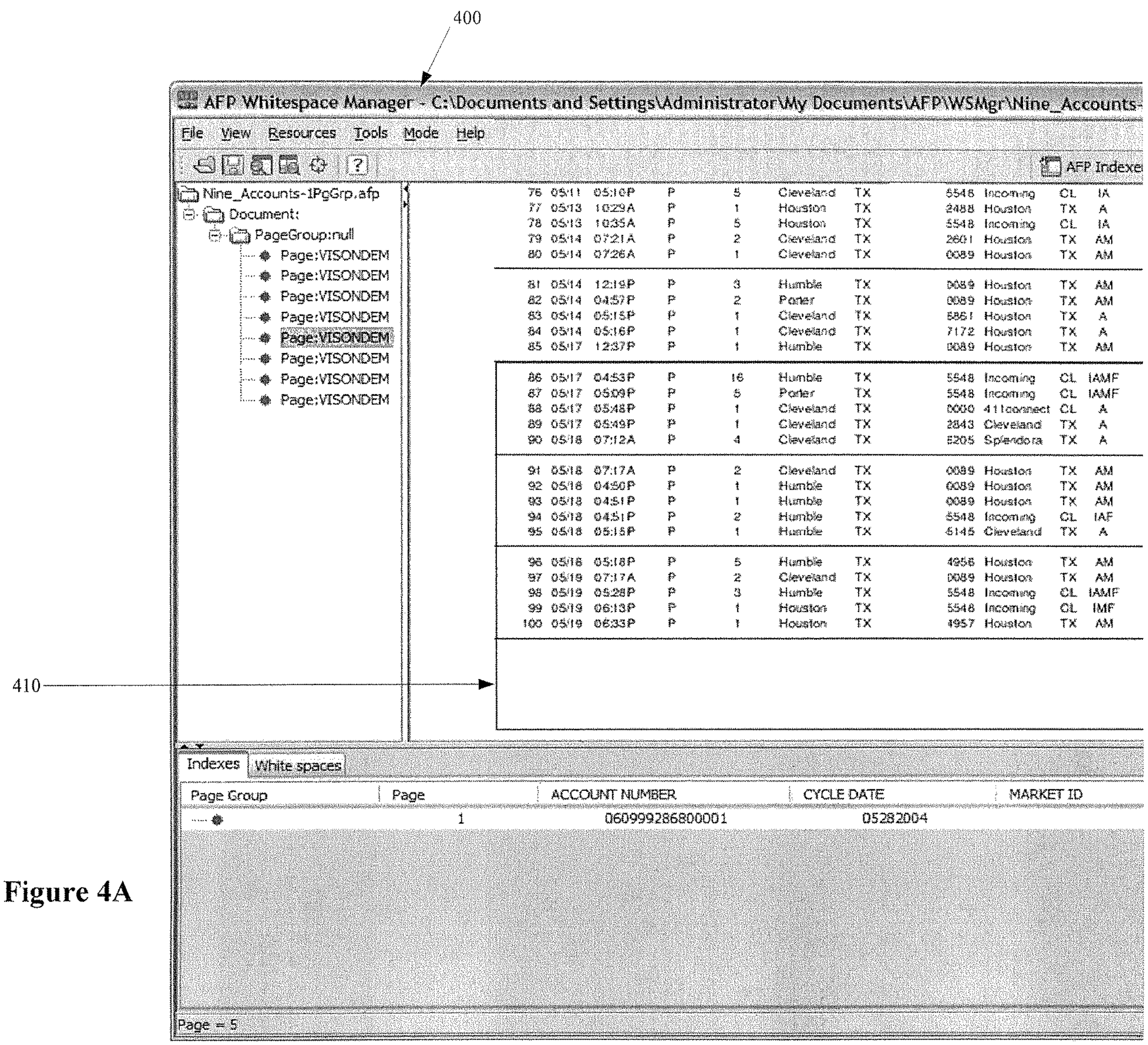

310
The quick brown fox jumped over the lazy brown dog ... poor sleepy pu
The quick brown fox jumped over the lazy brown dog ... poor sleepy pu
The quick brown fox jumped over the lazy brown dog ... poor sleepy pu
The quick brown fox jumped over the lazy brown dog ... poor sleepy pu
The quick brown fox jumped over the lazy brown dog ... poor sleepy pu
The quick brown fox jumped over the lazy brown dog ... poor sleepy pu
The quick brown fox jumped over the lazy brown dog ... poor sleepy pu
The quick brown fox jumped over the lazy brown dog ... poor sleepy pu
The quick brown fox jumped over the lazy brown dog ... poor sleepy pu
(LAST line of actual text within the search region...)
320
300
330

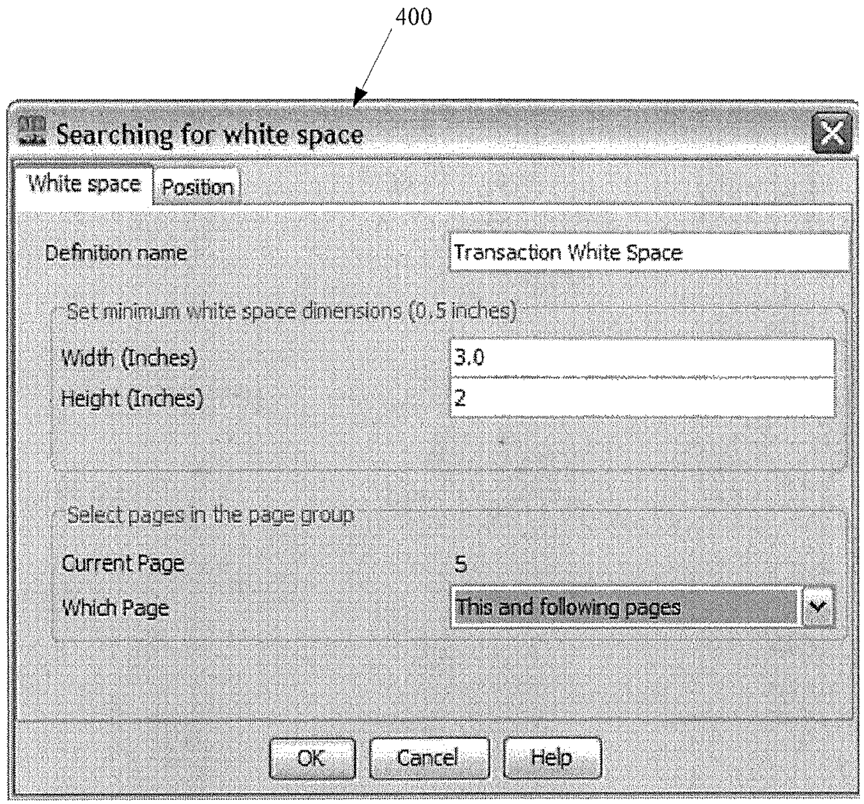
Figure 3



```
 76 05/11  05:10P   P    5   Cleveland  TX        5548 Incoming    CL  IA
 77 05/13  10:29A   P    1   Houston    TX        2488 Houston     TX  A
 78 05/13  10:35A   P    5   Houston    TX        5548 Incoming    CL  IA
 79 05/14  07:21A   P    2   Cleveland  TX        2601 Houston     TX  AM
 80 05/14  07:26A   P    1   Cleveland  TX        0089 Houston     TX  AM

 81 05/14  12:19P   P    3   Humble     TX        0089 Houston     TX  AM
 82 05/14  04:57P   P    2   Porter     TX        0089 Houston     TX  AM
 83 05/14  05:15P   P    1   Cleveland  TX        5861 Houston     TX  A
 84 05/14  05:16P   P    1   Cleveland  TX        7172 Houston     TX  A
 85 05/17  12:37P   P    1   Humble     TX        0089 Houston     TX  AM

 86 05/17  04:53P   P   16   Humble     TX        5548 Incoming    CL  IAMF
 87 05/17  05:09P   P    5   Porter     TX        5548 Incoming    CL  IAMF
 88 05/17  05:48P   P    1   Cleveland  TX        0000 411connect  CL  A
 89 05/17  05:49P   P    1   Cleveland  TX        2843 Cleveland   TX  A
 90 05/18  07:12A   P    4   Cleveland  TX        5205 Splendora   TX  A

 91 05/18  07:17A   P    2   Cleveland  TX        0089 Houston     TX  AM
 92 05/18  04:50P   P    1   Humble     TX        0089 Houston     TX  AM
 93 05/18  04:51P   P    1   Humble     TX        0089 Houston     TX  AM
 94 05/18  04:51P   P    2   Humble     TX        5548 Incoming    CL  IAF
 95 05/18  05:15P   P    1   Humble     TX        6145 Cleveland   TX  A

 96 05/18  05:18P   P    5   Humble     TX        4956 Houston     TX  AM
 97 05/19  07:17A   P    2   Cleveland  TX        0089 Houston     TX  AM
 98 05/19  05:28P   P    3   Humble     TX        5548 Incoming    CL  IAMF
 99 05/19  06:13P   P    1   Houston    TX        5548 Incoming    CL  IMF
100 05/19  06:33P   P    1   Houston    TX        4957 Houston     TX  AM
```

Page Group	Page	ACCOUNT NUMBER	CYCLE DATE	MARKET ID
●	1	060999286800001	05282004	

Figure 4A



Searching for white space

Definition name: Transaction White Space

Set minimum white space dimensions (0.5 inches)

Width (Inches): 3.0

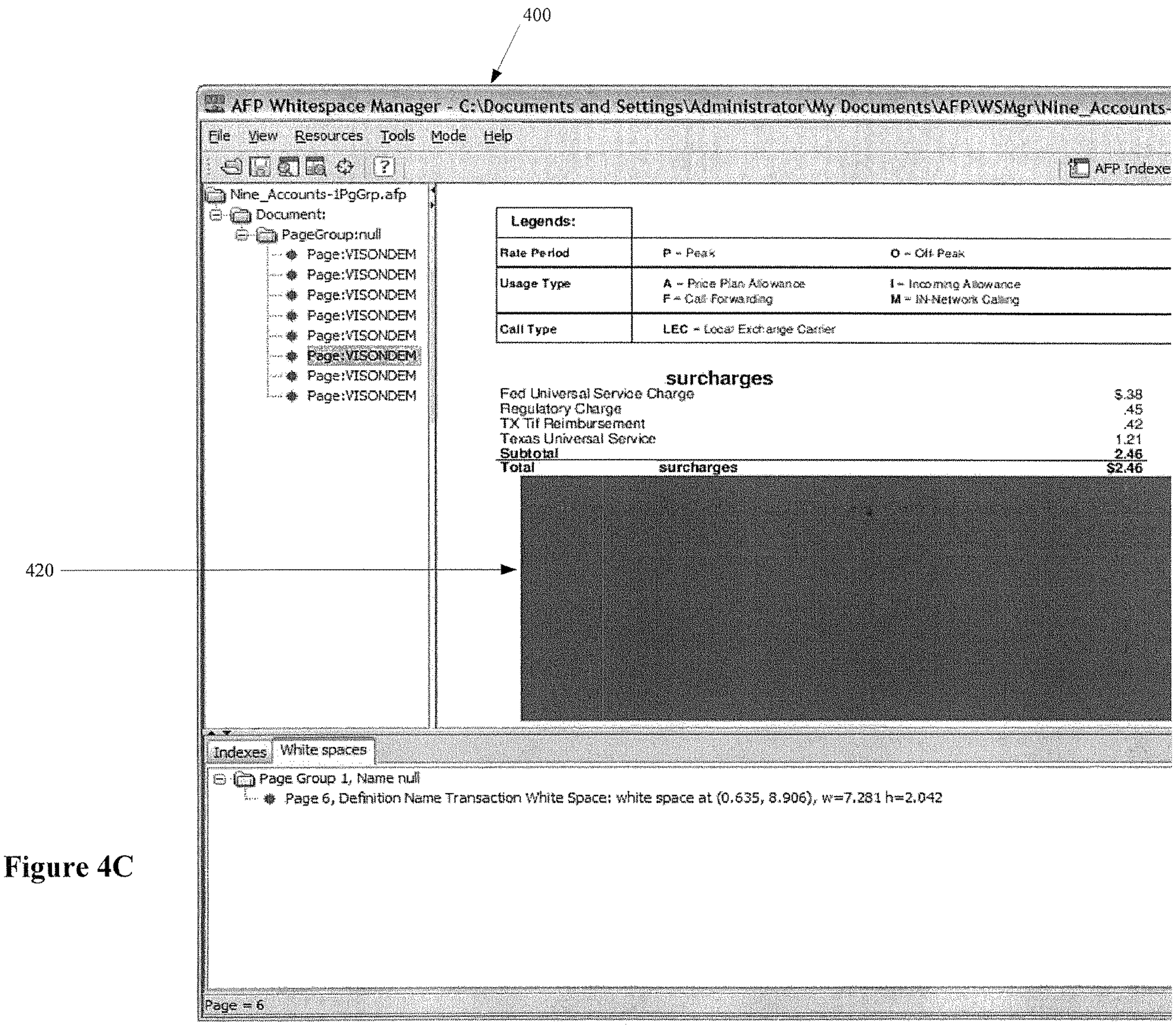
Height (Inches): 2

Select pages in the page group

Current Page: 5

Which Page: This and following pages

Figure 4B



Legends:		
Rate Period	P – Peak	O – Off Peak
Usage Type	A – Price Plan Allowance F – Call Forwarding	I – Incoming Allowance M – IN-Network Calling
Call Type	LEC – Local Exchange Carrier	

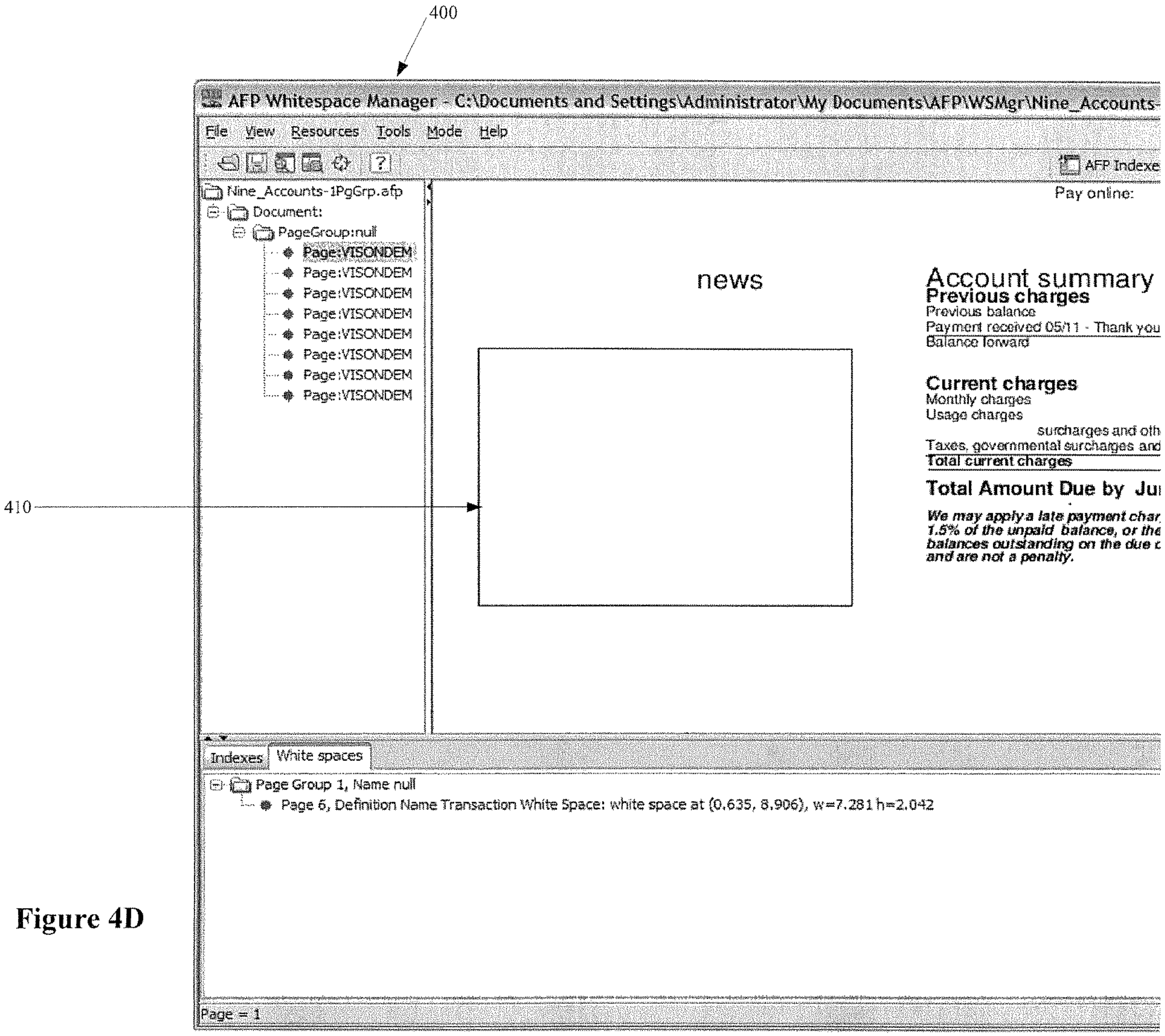
surcharges

Fed Universal Service Charge		$.38
Regulatory Charge		.45
TX Tif Reimbursement		.42
Texas Universal Service		1.21
Subtotal		**2.46**
Total	**surcharges**	**$2.46**

Page Group 1, Name null

- Page 6, Definition Name Transaction White Space: white space at (0.635, 8.906), w=7.281 h=2.042

Figure 4C



Pay online:

news

Account summary
Previous charges
Previous balance
Payment received 05/11 - Thank you
Balance forward

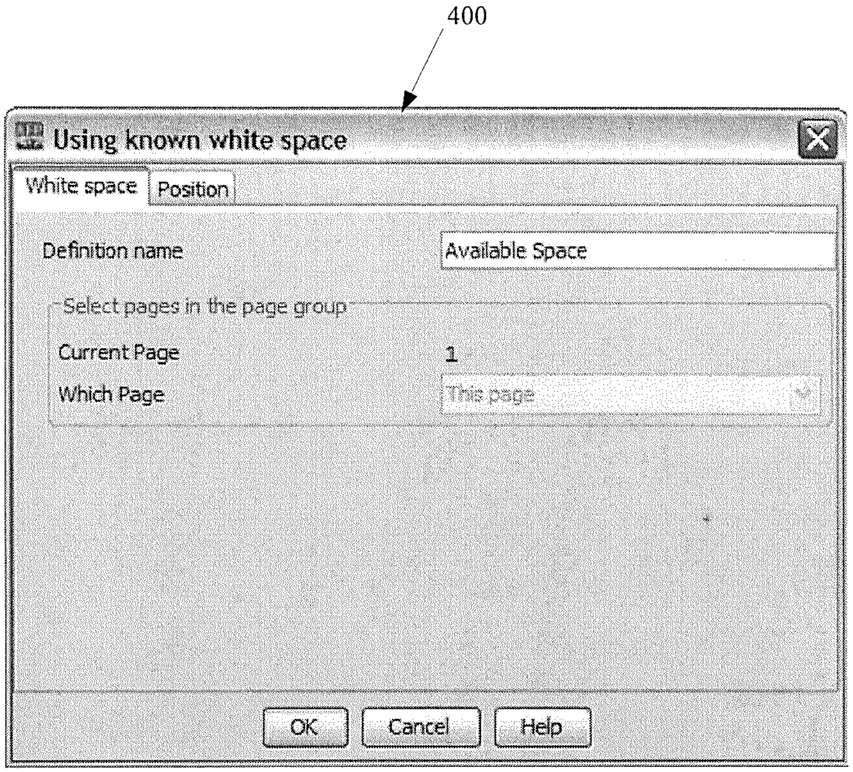
Current charges
Monthly charges
Usage charges
surcharges and oth
Taxes, governmental surcharges and
Total current charges

Total Amount Due by Ju

We may apply a late payment char 1.5% of the unpaid balance, or the balances outstanding on the due c and are not a penalty.

Page Group 1, Name null
- Page 6, Definition Name Transaction White Space: white space at (0.635, 8.906), w=7.281 h=2.042

Figure 4D



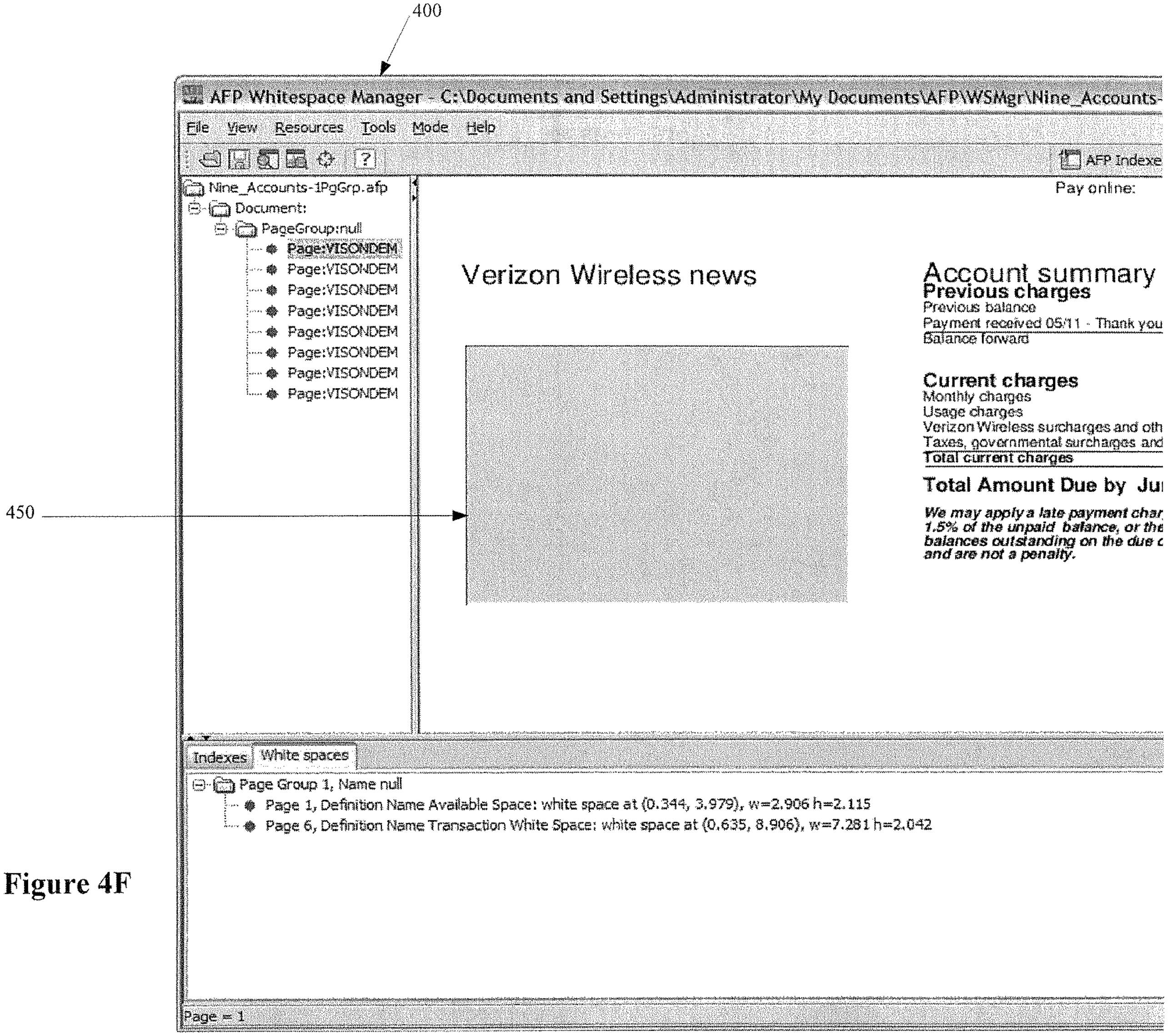
Definition name: Available Space

Select pages in the page group

Current Page: 1

Which Page: This page

Figure 4E


400
450
AFP Whitespace Manager - C:\Documents and Settings\Administrator\My Documents\AFP\WSMgr\Nine_Accounts-
File View Resources Tools Mode Help
?
AFP Indexe
Indexes
White spaces
Page = 1

Nine_Accounts-1PgGrp.afp
⊟ Document:
⊟ PageGroup:null
- ● Page:VISONDEM
- ● Page:VISONDEM
- ● Page:VISONDEM
- ● Page:VISONDEM
- ● Page:VISONDEM
- ● Page:VISONDEM
- ● Page:VISONDEM
- ● Page:VISONDEM

Pay online:

Verizon Wireless news

Account summary
Previous charges
Previous balance
Payment received 05/11 - Thank you
Balance forward

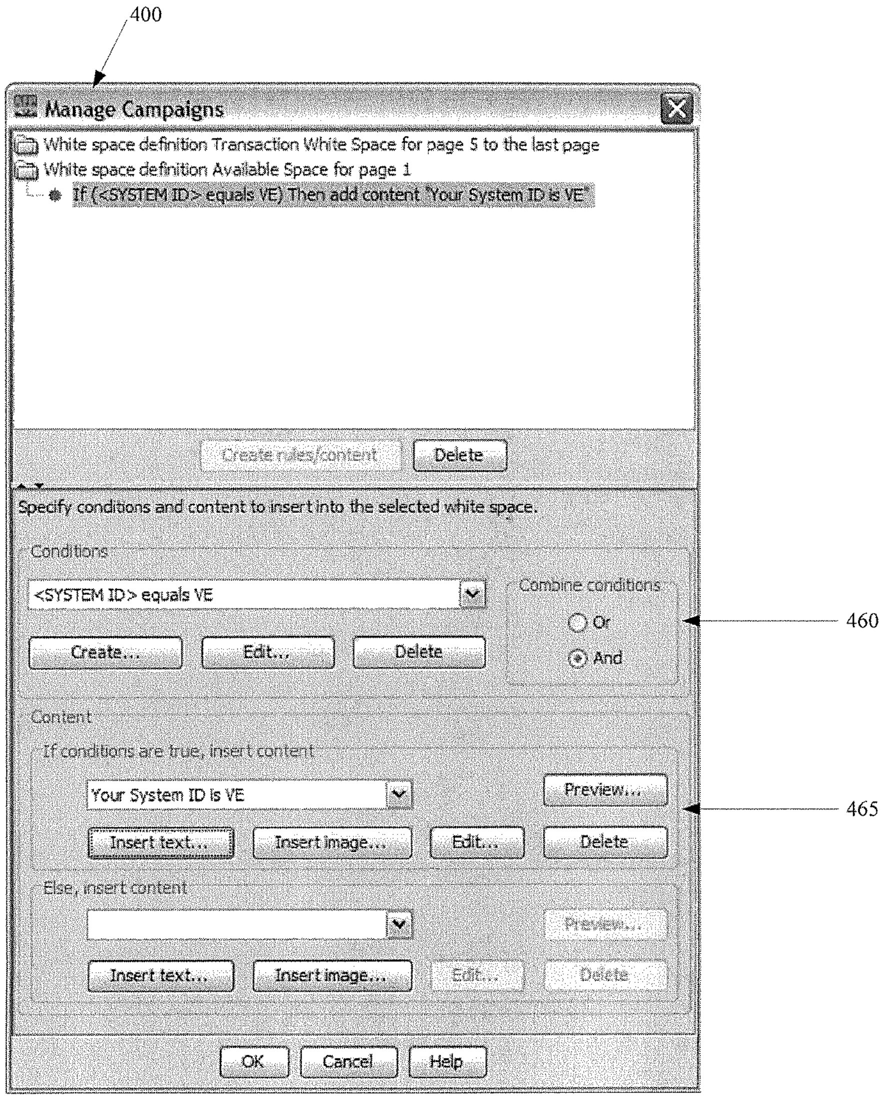
Current charges
Monthly charges
Usage charges
Verizon Wireless surcharges and oth
Taxes, governmental surcharges and
Total current charges

Total Amount Due by Ju

We may apply a late payment char 1.5% of the unpaid balance, or the balances outstanding on the due c and are not a penalty.

⊟ Page Group 1, Name null
- ● Page 1, Definition Name Available Space: white space at (0.344, 3.979), w=2.906 h=2.115
- ● Page 6, Definition Name Transaction White Space: white space at (0.635, 8.906), w=7.281 h=2.042

Figure 4F



Manage Campaigns

- White space definition Transaction White Space for page 5 to the last page
- White space definition Available Space for page 1
  - If (<SYSTEM ID> equals VE) Then add content "Your System ID is VE"

Specify conditions and content to insert into the selected white space.

Conditions

<SYSTEM ID> equals VE

Combine conditions

○ Or ⦿ And

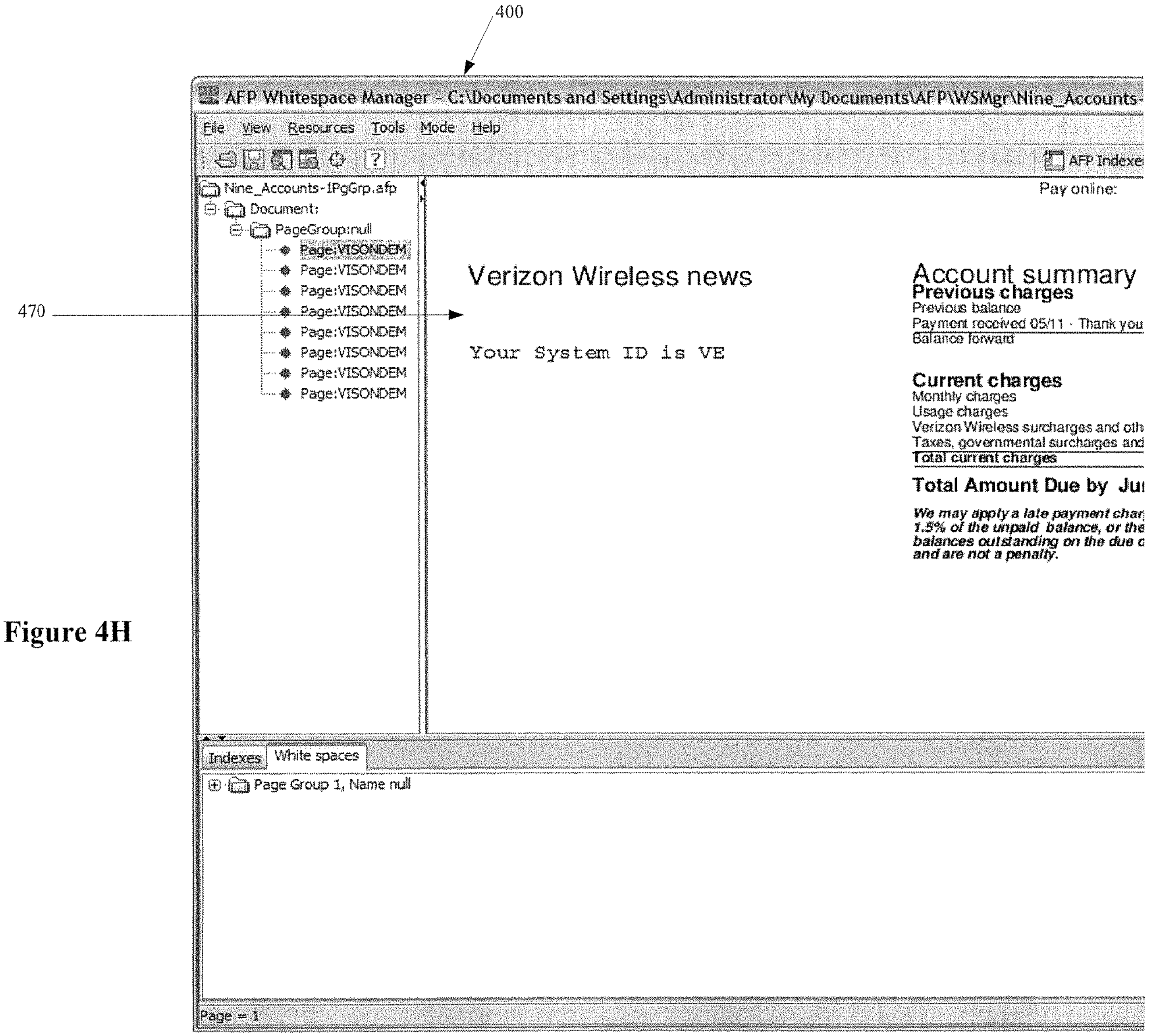
Content

If conditions are true, insert content

Your System ID is VE

Else, insert content

Figure 4G



Pay online:

Verizon Wireless news

Your System ID is VE

Account summary
Previous charges
Previous balance
Payment received 05/11 - Thank you
Balance forward

Current charges
Monthly charges
Usage charges
Verizon Wireless surcharges and oth
Taxes, governmental surcharges and
Total current charges

Total Amount Due by Ju

We may apply a late payment char
1.5% of the unpaid balance, or the
balances outstanding on the due c
and are not a penalty.

Figure 4H

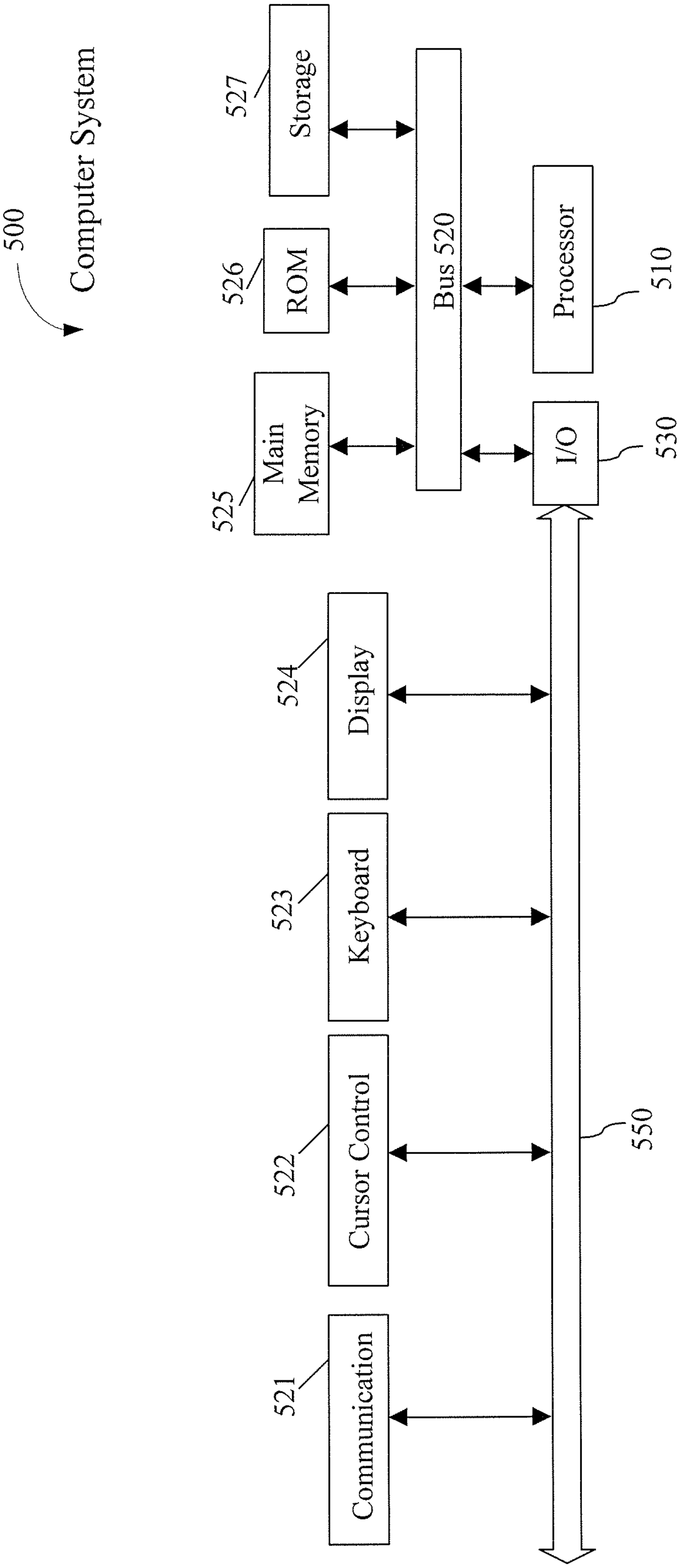

500
Computer System
527
Storage
526
ROM
525
Main Memory
Bus 520
Processor
510
I/O
530
524
Display
523
Keyboard
522
Cursor Control
521
Communication
550

Figure 5

WHITE SPACE MANAGEMENT MECHANISM

The present patent application is a continuation application claiming priority from currently application Ser. No. 15/665,087, filed Jul. 31, 2017, which is a continuation of application Ser. No. 12/771,005, filed Apr. 30, 2010, now U.S. Pat. No. 9,760,328.

FIELD OF THE INVENTION

The invention relates to the field of printing systems. Particularly, the invention relates to identifying and utilizing blank spaces in post-production print jobs.

BACKGROUND

Printers are common peripheral devices attached to computers. A printer allows a computer user to make a hard copy of documents that are created in a variety of applications and programs on a computer. To function properly, a channel of communication is established (e.g., via a network connection) between the printer and the computer to enable the printer to receive commands and information from the host computer.

Once a connection is established between a workstation and the printer, printing software is implemented at a print server to manage a print job from order entry and management through the complete printing process. The printing software may simultaneously manage in excess of thousands of print jobs that have been spooled (or queued) for production. Each of these print jobs may include various documents that include unused areas (or white spaces) at which content may be placed, but are left blank as a result of the defined print job data.

In many print applications, the existence of such magnitudes of white space is wasteful. For example, a print job performed by a bank that produces thousands of statements to be mailed to customers may result in hundreds of pages worth of white space. Thus, the bank is unable to maximize the efficiency of print jobs by utilizing unused space on print pages. Since the cost of producing the statements has already been expended at the time of production, the bank could take full advantage of such an investment by having an opportunity to utilize the white space to place content (e.g., advertisements, offers, etc.) for which a consumer may have interest.

Accordingly, a mechanism to identify and utilize white spaces in print jobs is desired.

SUMMARY

In one embodiment, a computer generated method is disclosed. The method includes examining a print job data stream spooled in a spooled print production file, identifying one or more white space regions within documents in the print production file and inserting content into each of the white space regions.

In another embodiment, a print server is disclosed. The print server includes a printing software product having a whitespace manager to examine a print job data stream spooled in a spooled print production file stored at the server, identify one or more white space regions within documents in the print production file and insert content into each of the white space regions.

In a further embodiment, a computer system is disclosed. The computer system includes a memory to store a printing software product, a processor, coupled to the memory, to execute the printing software product and a display device to display a graphical user interface (GUI) upon the processor executing the printing product. The GUI includes menu options to enable identification of one or more white space regions within documents in a print production file and selection of content to be inserted in the white space regions.

BRIEF DESCRIPTION OF THE DRAWINGS

A better understanding of the present invention can be obtained from the following detailed description in conjunction with the following drawings, in which:

FIG. **1** illustrates one embodiment of a data processing system network;

FIG. **2** is a flow diagram illustrating one embodiment of performing white space management;

FIG. **3** illustrates one embodiment of searching for a white space;

FIGS. **4**A-**4**H are screen shots of various embodiments of a GUI; and

FIG. **5** illustrates one embodiment of a computer system.

DETAILED DESCRIPTION

A whitespace management mechanism is described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without some of these specific details. In other instances, well-known structures and devices are shown in block diagram form to avoid obscuring the underlying principles of the present invention.

Reference in the specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment.

FIG. **1** illustrates one embodiment of a data processing system network **100**. Network **100** includes a data processing system **102**, which may be either a desktop or a mobile data processing system, coupled via communications link **104** to network **106**. In one embodiment, data processing system **102** is a conventional data processing system including a processor, local memory, nonvolatile storage, and input/output devices such as a keyboard, mouse, trackball, and the like, all in accordance with the known art. In one embodiment, data processing system **102** includes and employs the Windows operating system, or other operating system, and/or network drivers permitting data processing system **102** to communicate with network **106** for the purposes of employing resources within network **106**.

Network **106** may be a local area network (LAN) or any other network over which print requests may be submitted to a remote printer or print server. Communications link **104** may be in the form of a network adapter, docking station, or the like, and supports communications between data processing system **102** and network **106** employing a network communications protocol such as Ethernet, the AS/400 Network, or the like.

According to one embodiment, network **106** includes a print server **108** that serves print requests over network **106** received via communications link **110** between print server **108** and network **106**. Print server **108** subsequently transmits the print requests via communications link **110** to one of printers **109** for printing, which are coupled to network **106** via communications links **111**.

In one embodiment, a print application at data processing system **102** allows a user to select the desired print server **108** and submit requests for service requests to printer **109** via print server **108** over network **106**. In a further embodiment, the print application implements the (Advanced Function Presentation) AFP™ presentation system developed by International Business Machines Corporation to represent documents in a data format that is independent of the methods that are utilized to capture or create those documents.

According to the AFP system, documents may include combinations of text, image, graphics, and/or barcode objects in device and resolution independent formats. Documents may also include and/or reference fonts, overlays, and other resource objects, which are required at presentation time to present the data properly. In other embodiments, additional/alternative presentation architectures may be implemented at the print application.

However in an AFP embodiment, the print application provides a Mixed Object Document Content Architecture (MO:DCA) data stream to print server **108**. In such an embodiment, the AFP MO:DCA data streams are object-oriented streams including, among other things, data objects, page objects, and resource objects. In a further embodiment, AFP MO:DCA data streams include a Resource Environment Group (REG) that is specified at the beginning of the AFP document, before the first page. When the AFP MO:DCA data streams are processed by print server **108**, the REG structure is encountered first and causes server **108** to download any of the identified resources that are not already present in the printer.

Although described as separate entities, other embodiments may include print server **108** being incorporated in one or more of the printers **109**. In yet further embodiments, the print server and printer may be physically separate entities. Therefore, the data processing system network depicted in FIG. **1** is selected for the purposes of explaining and illustrating the present invention and is not intended to imply architectural limitations. Those skilled in the art will recognize that various additional components may be utilized in conjunction with the present invention.

According to one embodiment, print server **108** implements a printing software product that manages the printing of documents from data processing system **102** and one or more of printers **109**. In other embodiments, the printing software product manages printing of documents from multiple data processing systems **102** to the one or more printers **109**. In one embodiment, the printing software product may be implemented using either InfoPrint Manager (IPM) or InfoPrint ProcessDirector (IPPD), although other types of printing software may be used instead. In a further embodiment, the print application at data processing system **102** interacts with the printing software product to provide for efficient transmission of print jobs.

In one embodiment, the printing software product includes a graphical user interface (GUI) **120** accessible at a data processing system **102** that enables a system administrator (or operator) to interact with the printing software product and print application. In such an embodiment, GUI **120** includes a whitespace manager that records and manages white space identified in documents spooled for production at the printing software product.

FIG. **2** is a flow diagram illustrating one embodiment of a process performed on a document by a whitespace manager. At processing block **210**, the whitespace manager identifies one or more white spaces on a print job spooled at the printing software product. In one embodiment, three types of white spaces may be identified. These include BLANK pages, FIXED spaces and SEARCH spaces. As will be described in greater detail below, the fixed and search spaces may involve operator interaction with the whitespace manager in order to identify regions of white space on a page.

At processing block **220**, the identified white spaces are quantified. For instance, fixed white space regions are quantified in terms of the size and type of (e.g., text or image) content that may be placed within the white space. This quantification may subsequently be codified and stored. In one embodiment, a quantification value may be a number of characters of message text in a certain point size that would fit in the white space region, while in other embodiments it may include dimensions of an image that could fit within the region.

At processing block **230**, one or more white space regions are filled with content. In one embodiment, a white space region may be filled with a predetermined message. However in other embodiments, dynamic messages may be used to fill a region based on variable data included on a document. For example, whitespace manager may analyze financial data included on a bank statement that is waiting to be printed. Based on the analysis whitespace manager may fill the white space region with content appropriate to a financial message corresponding to the analysis.

As discussed above, whitespace manager identifies BLANK pages, FIXED spaces and SEARCH spaces. Completely blank pages are one of the greatest sources of white space in current print applications. For instance, most duplex (e.g., N-up duplex) applications force the end of one client's statement in a manner that precludes any possibility of inadvertent exposure of the statement to another client.

Within AFP this process is implemented by an Eject to Front Facing (EFF). However, the process may also be implemented by adding actual blank pages to the application (e.g., padding any odd-pg statement to an even page count). A minor variant to the blank page is the "almost" blank page, which appears in a datastream, but provides no useful message content (e.g., "this page intentionally left blank" pages). Because of blank pages attributed to EFF and similar processes, a 25% increase of actual pages printed from pages of a print job file may occur. For example, a client application that includes all three page long statements may result in a result in a 75,000 page print file becoming-100,000 pages at print time.

Therefore, the whitespace manager automatically identifies spaces where a blank page exists and fills the pages based upon one or more predetermined rules for indicating how the pages are to be filled with appropriate messages. According to one embodiment, the whitespace manager examines a print job data stream. In such an embodiment, the whitespace manager uses page data, mail piece breaks, duplex/simplex, and any advanced settings (e.g., constant back IMM) to identify the blank pages.

In embodiments using 2-up duplex (e.g., pages scaled to 2 pages per page using double-sided printing), odd-page-length documents are padded with a real page at the end of the document to preclude ejecting blank pages when the printer is handling the data. Thus, the whitespace manager normalizes all data streams to include an even number of page images so that each physical sheet of paper is filled front and back before being sent to the printer.

In other embodiments, pages are identified in existing legacy streams, such as "this page intentionally left blank" pages). Thus, blank pages may be identified by identification of a small area (e.g., the region in which that legacy text appears). If a page is found to be blank other than the specific region(s) identified, the page is deemed to be a "blank" page, and is swapped out in favor of a targeted message page. Other embodiments may include looking for a particular string on a page such (e.g., "this page intentionally left blank").

Once identified, blank pages are filled with a real page of messages based on user defined rules. In one embodiment, a message may include an entire pre-defined page that includes a static set of messages of interest to many clients. For example, a bank may add a simple 8.5×11 page that identifies current mortgage rates, CD rates, a website to visit, and/or other messages of potential interest.

In another embodiment, the identified blank page may be divided into a series of targeted messages including smaller coupons, messages, etc. In yet a further embodiment, the blank page may be filled with targeted trans-promotional messages tailored to a specific client, taking into account variable data included in the document. For instance, the variable data may include geographic region, value of the statement, and client preferences.

For FIXED space management, the whitespace manager is utilized by an operator to identify a fixed region on a page. In such an embodiment, the selected region is one that is known to be blank. Fixed space regions are often rectangular blank spaces within the top or bottom area of a page. However in further embodiments, other types of shaped regions may be identified and managed in a similar way (e.g., a circle could identify a suitable region for a circular message (image or text) to be placed on the page).

According to one embodiment, the operator identifies the region as suitable for white space message placement. In such an embodiment, identification of a region may be implemented using any reasonable method (e.g., Cartesian or Polar coordinates, etc.). However, in other embodiments identification is implemented using an interactive system against a visual image of a production file in order to identify a precise region to be managed as fixed white space. Within the whitespace manager this is done using a GUI and a mouse, against a visual representation of a page data for a particular application.

Once identified, the fixed regions are quantified in terms of the size of messages (text or image) that can be placed within them. As discussed above, the quantified data is then codified and stored. These stored quantified regions are subsequently combined with run-time business rules to enable the whitespace manager to fill the regions. Like for BLANK space management, a fixed region's dimensions are known in advance, which allows for messages to be tailored in advance to the specific size of the fixed spaces.

In one embodiment, a safety mechanism is constructed at run time. The mechanism checks the fixed space against the actual page data to make sure the region is actually blank. The mechanism may be performed in absolute terms by checking actual rendered images of page data, or in relative terms by allowing for underlying forms and logos via an overlay, to confirm that no text or foreground images occupy the fixed space region that is supposed to be blank. In a further embodiment, the safety mechanism includes error handling to provide an alert that the region is not blank. The error handling may include, for example, a simple warning message and suppression of an individual message placement, or raising an error condition since the region was identified by the operator as one that was supposed to be completely blank.

SEARCH space management is implemented in applications for which there is a region of a page that can be completely full, but is often only partially filled. This is common, for example, in billing applications where a particular region of a document is filled with bank transactions, stock trades, phone records, or other detailed line item data. This data is typically reported via one or more transactional pages filled with the billing details.

However, the last of the transactional pages, depending upon how precisely the actual transaction details fill the available space, often has some variable amount of blank space. Particularly, the last page usually includes a random distribution of 1-N transactions, thus leaving from zero to (N-1) rows of whitespace, where N is the maximum number transactions that may listed on the page.

In other instances, large regions of a page may be intended for text, but may not be entirely filled via an application that produces the data, thus leaving a white space. According to one embodiment, the whitespace manager identifies such white space in order to fill the space with appropriate messages.

As with the fixed spaces, SEARCH space management begins with an operator utilizing the whitespace manager to identify each region in a document that may include white space. In one embodiment, a region definition is checked against a sample/reference file to report a number of pages that include actual white space in a particular region, as well as the sizes of the white space. Thus, the whitespace manager enables interactive planning and assessment of a live production file for white space available within a certain region.

In a further embodiment, a rectangle is drawn around an entire region to be searched (e.g., a text portion of an automated letter application, a transactional data area, etc.) in order to identify a search region. As discussed above, other shaped regions types (circle, triangle, etc.) may be identified and managed. Subsequently, at run-time, the white space manager identifies an appropriate subset of the search space that would be available as white space for message placement.

As an example, if an identified 5.5"×4" area is found to have text in the top 1" only, an appropriate subset (e.g., 4.5"×4") will be identified as white-space. Such identification can also allow for margins, kerning, and other details. In this example, the region identified may be only 4"×4" if allowing for a 0.5" margin was desired to set whitespace messages apart. FIG. **3** illustrates one embodiment of searching for a white space on a page **300**. As shown in FIG. **3**, page **300** includes a search area **310** with margins **320**. The whitespace manager searches page **300** until identifying white space **330** in search area **310**, between margins **320**.

Also during run time, search spaces are checked against the actual data within the file to assess if criteria for message placement are met, and if so, the regions may be filled with the desired message. As discussed above, business rules are used during run time to guide how each space is filled. In one embodiment a region is checked, during run time evaluation, against one or more bitmap planes of page images (e.g., in the CMYK (or color) or K (or black/white) planes) to make sure that no text or image intrudes upon the space.

However in other embodiments, text or image messages may be placed on top of an overlay or other type of "image" even if some background pixels occur within the search space. Therefore, the whitespace manager may look for page text that lands within a region at a current time. In a further embodiment, the whitespace manager stores an image of all documents for which white space has been filled in order to include a precise record of the appearance of messages placed on each document.

FIGS. **4**A-**4**H are screen shots of various embodiments of a GUI showing a GUI **120** AFP whitespace manager **400**. FIG. **4**A illustrates a screen shot of one embodiment whitespace manager **400** having a region **410** that is selected in a transactional page of phone billing details to search in subsequent pages of a document. FIG. **4**B illustrates a screen shot of another embodiment of whitespace manager **400** which enables an operator to define rules for the white space. For instance, an operator may set minimum white space dimensions to search for, as well as select which pages to search. FIG. **4**C illustrates a screen shot of an embodiment of whitespace manager **400** showing the results of the search. As shown in FIG. **4**C, a region **420** matching the criteria of the search has been found. Further, a bottom pane of whitespace manager **400** provides the details of the white space region.

FIG. **4**D illustrates a screen shot of one embodiment whitespace manager **400** in which region **410** is a known white space region. FIG. **4**E illustrates a screen shot of an embodiment of the rules definition page of whitespace manager **400** corresponding to the selection of the known white space region. FIG. **4**F illustrates a screen shot of an embodiment of whitespace manager **400** showing a region **450** resulting from search results from selecting a known region selection. Further, the bottom pane of whitespace manager **400** provides a list, along with the details of the found white space regions.

FIG. **4**G illustrates a screen shot of one embodiment whitespace manager **400** used for specifying rules for adding content to the found white space regions. As shown in FIG. **4**G, whitespace manager **400** includes conditions section **460** and content section **465**, each including various buttons and menus to establish conditions and content for filling white space regions. For instance, an operator may enter text to be inserted to a whitespace region upon conditions entered in section **460** being met.

According to one embodiment, each whitespace region has its own set of rules. FIG. **4**H illustrates a screen shot of an embodiment of whitespace manager **400** showing region **450** being filled with text **470** resulting from the text entered in content section **465**.

FIG. **5** illustrates a computer system **500** on which data processing system **102** and/or server **108** may be implemented. Computer system **500** includes a system bus **520** for communicating information, and a processor **510** coupled to bus **520** for processing information.

Computer system **500** further comprises a random access memory (RAM) or other dynamic storage device **525** (referred to herein as main memory), coupled to bus **520** for storing information and instructions to be executed by processor **510**. Main memory **525** also may be used for storing temporary variables or other intermediate information during execution of instructions by processor **510**. Computer system **500** also may include a read only memory (ROM) and or other static storage device **526** coupled to bus **520** for storing static information and instructions used by processor **510**.

A data storage device **527** such as a magnetic disk or optical disc and its corresponding drive may also be coupled to computer system **500** for storing information and instructions. Computer system **500** can also be coupled to a second I/O bus **550** via an I/O interface **530**. A plurality of I/O

devices may be coupled to I/O bus **550**, including a display device **524**, an input device (e.g., an alphanumeric input device **523** and or a cursor control device **522**). The communication device **521** is for accessing other computers (servers or clients). The communication device **521** may comprise a modem, a network interface card, or other well-known interface device, such as those used for coupling to Ethernet, token ring, or other types of networks.

Embodiments of the invention may include various steps asset forth above. The steps may be embodied in machine-executable instructions. The instructions can be used to cause a general-purpose or special-purpose processor to perform certain steps. Alternatively, these steps may be performed by specific hardware components that contain hardwired logic for performing the steps, or by any combination of programmed computer components and custom hardware components.

Elements of the present invention may also be provided as a machine-readable medium for storing the machine-executable instructions. The machine-readable medium may include, but is not limited to, floppy diskettes, optical disks, CD-ROMs, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnetic or optical cards, propagation media or other type of media/machine-readable medium suitable for storing electronic instructions. For example, the present invention may be downloaded as a computer program which may be transferred from a remote computer (e.g., a server) to a requesting computer (e.g., a client) by way of data signals embodied in a carrier wave or other propagation medium via a communication link (e.g., a modem or network connection).

Whereas many alterations and modifications of the present invention will no doubt become apparent to a person of ordinary skill in the art after having read the foregoing description, it is to be understood that any particular embodiment shown and described by way of illustration is in no way intended to be considered limiting. Therefore, references to details of various embodiments are not intended to limit the scope of the claims, which in themselves recite only those features regarded as essential to the invention.

What is claimed is:

1. A printing system comprising:

a memory to store a printing software product; and

a processor to execute the printing software product to cause the printing software product to examine a print job data stream spooled in a spooled print production file, identify one or more blank pages within documents in the print production file and insert content into each of the blank pages based on one or more rules associated with each blank page, wherein the rules identify one or more dynamic messages that are to be inserted into a blank page based upon variable data included in the print job data stream.

2. The printing system of claim **1**, wherein a blank page is identified based on attributes in the print job data stream.

3. The printing system of claim **2**, wherein identifying a blank page in the print job data stream comprises identifying attributes in the print job data stream indicating odd page length documents in a n-up duplex format.

4. The printing system of claim **2**, wherein identifying a blank page in the print job data stream comprises identifying one or more text object attributes in the print job data stream indicating that the page is to be left blank.

5. The printing system of claim **1**, wherein the one or more rules indicates that the blank page is to be filled with a single message.

6. The printing system of claim **1**, wherein the one or more rules indicates that the blank page is to be divided into a plurality of messages.

7. The printing system of claim **1**, wherein the variable data comprises on or more of: geographic data, user preferences and a statement value.

8. A printing system comprising:

a memory to store a printing software product; and

a processor to execute the printing software product to cause the printing software product to examine a print job data stream spooled in a spooled print production file, identify one or more fixed white space regions within documents in the print production file, quantify each of the fixed white space regions to establish one or more characteristics of the content to be inserted into each fixed white space region and insert content into each of the fixed white space region based on one or more rules associated with each fixed white space region, wherein the rules identify one or more dynamic messages that are to be inserted into a fixed white space region based upon variable data included in the print job data stream.

9. The printing system of claim **8**, wherein identifying the fixed white space region comprises comparing the fixed white space region to page data to determine if the fixed white space region is blank.

10. The printing system of claim **9**, wherein the printing software product generates an alert upon a determination that the fixed white space region is not blank.

11. The printing system of claim **8**, wherein the fixed white space region is identified using coordinates.

12. The printing system of claim **8**, wherein the fixed white space region is identified using a graphical user interface to access an image of the print production file.

13. A printing system comprising:

a memory to store a printing software product; and

a processor to execute the printing software product to cause the printing software product to examine a print job data stream spooled in a spooled print production file, identify one or more search white space regions within documents in the print production file, quantify each of the search white space regions to establish one or more characteristics of the content to be inserted into each search white space region and insert content into each of the search white space region based on one or more rules associated with each search white space region, wherein the rules identify one or more dynamic messages that are to be inserted into a search white space region based upon variable data included in the print job data stream.

14. The printing system of claim **13** a search white space region comprises a region in one or more documents in the print production file that is identified by searching regions in documents that have variable white space regions.

15. The printing system of claim **14**, wherein identifying a search white space region comprises checking a region definition against a file to report pages that include white space in a region.

16. The printing system of claim **15**, wherein identifying the search white space region further comprises:

selecting a search region;

identifying a subset of the search region available for content placement; and

comparing the subset of the search region to the print production file to determine if content placement is available.

17. The printing system of claim **16**, wherein the search white space region is identified using a graphical user interface to access an image of the print production file.

18. The printing system of claim **17**, wherein selecting the search region comprises using the graphical user interface to draw a shape around the subset of the search region.

19. The printing system of claim **13**, wherein the printing software product compares the one or more search white space regions with file data to determine if criteria for inserting the content has been met.

20. The printing system of claim **19**, wherein the printing software product compares a search white space region against one or more bitmap planes of page images to determine whether objects intrude into the search white space region.

* * * * *